

Utilizing the SAS® Macro Facility to Interface with PROC COMPARE

Robert Adams, Merck & Co., Inc., North Wales, PA

ABSTRACT

SAS® is used in a variety of ways in pharmaceutical drug development, including to validate the data sets from which tables, figures, and listings are derived. The SAS procedure PROC COMPARE is useful when validating data sets since it can compare the contents of two data sets, selected variables in different data sets, or variables within the same data set. In addition, the procedure uses an automatic macro variable to store a return code that provides information about the result of the comparison. This paper will show how to use the value of the macro variable populated by PROC COMPARE to conditionally execute parts of a SAS program based on the result of the comparison.

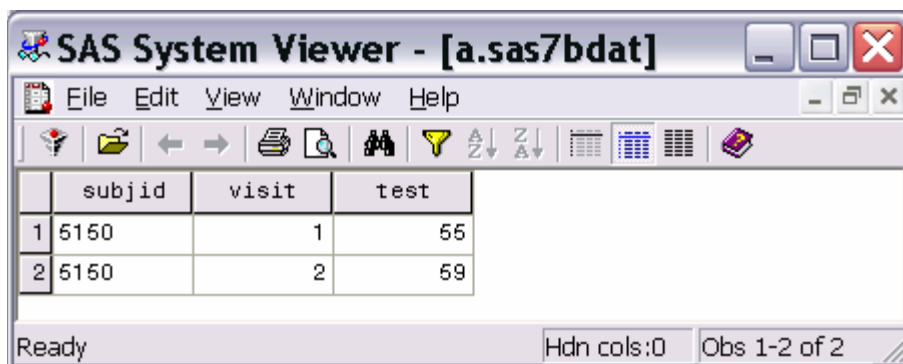
KEYWORDS

Automatic macro variable, procedure return code

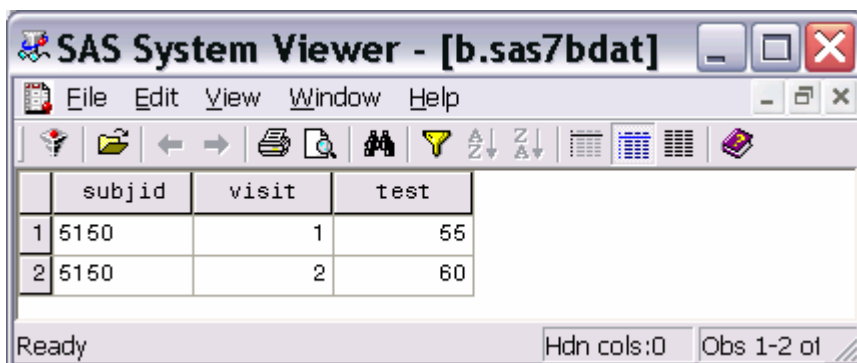
INTRODUCTION

One of the ways that PROC COMPARE reports the results of its comparisons is by storing a return code in the automatic macro variable SYSINFO. The value of the return code provides information about the results of the comparison. By checking the value of SYSINFO after PROC COMPARE has run and before any other step begins, subsequent SAS steps can use the results of a PROC COMPARE step to determine what action to take or what parts of a SAS program to execute.

EXAMPLE DATA SETS



	subjid	visit	test
1	5150	1	55
2	5150	2	59



	subjid	visit	test
1	5150	1	55
2	5150	2	60

MACRO RETURN CODES

The possible values for the SYSINFO variable after PROC COMPARE has run are listed below. Keep in mind that the value of SYSINFO must be obtained before another SAS step starts because every SAS step resets SYSINFO.

<i>Macro Return Codes</i>				
Bit	Condition	Code	Hex	Description
1	DSLABEL	1	0001X	Data set labels differ
2	DSTYPE	2	0002X	Data set types differ
3	INFORMAT	4	0004X	Variable has different informat
4	FORMAT	8	0008X	Variable has different format
5	LENGTH	16	0010X	Variable has different length
6	LABEL	32	0020X	Variable has different label
7	BASEOBS	64	0040X	Base data set has observation not in comparison
8	COMPOBS	128	0080X	Comparison data set has observation not in base
9	BASEBY	256	0100X	Base data set has BY group not in comparison
10	COMPBY	512	0200X	Comparison data set has BY group not in base
11	BASEVAR	1024	0400X	Base data set has variable not in comparison
12	COMPVAR	2048	0800X	Comparison data set has variable not in base
13	VALUE	4096	1000X	A value comparison was unequal
14	TYPE	8192	2000X	Conflicting variable types
15	BYVAR	16384	4000X	BY variables do not match
16	ERROR	32768	8000X	Fatal error: comparison not done

Notice that the return codes are scaled to enable a simple check of the degree to which the data sets differ.

CODING TO USE MACRO RETURN CODES

To conditionally execute a SAS program based on the result of PROC COMPARE, the procedure must run, the value of the SYSINFO macro variable must be obtained, then conditional logic can be applied. For example, to check that two data sets contain the same variables, observations, and values without concern for differences in labels, formats, etc. the following statements can be used:

```

proc compare base = files.a compare = files.b ;
run ;

%let rc = &sysinfo ;

data _null_ ;
    if &rc ge 64 then do ;
        put 'WARNING: The number of observations, BY groups, or
            variables are unequal' ;
        put 'or the variable values, variable types, or BY
            variables do not match' ;
    end ;
run ;

```

LOG OUTPUT

```

628  proc compare base = files.a compare = files.b ;
629  run ;

```

NOTE: There were 2 observations read from the data set FILES.A.

NOTE: There were 2 observations read from the data set FILES.B.

NOTE: PROCEDURE COMPARE used (Total process time):

```

    real time          0.01 seconds
    cpu time           0.01 seconds

```

```

630
631  %let rc = &sysinfo ;
632
633  data _null_ ;
634  if &rc ge 64 then
635      do ;
636          put 'WARNING: The number of observations, BY groups, or variables are
              unequal' ;
637          put 'or the variable values, variable types, or BY variables do not match';
638      end ;
639  run ;

```

WARNING: The number of observations, BY groups, or variables are unequal
or the variable values, variable types, or BY variables do not match

APPLICATIONS

Obtaining the return code from the automatic macro variable populated by PROC COMPARE lends itself very easily to the development of a macro that can handle the different scenarios that result from the comparison. For example, a programming group frequently validating data sets using PROC COMPARE could develop a macro that can take the return code from PROC COMPARE as a parameter and perform error handling or provide simple feedback instead of the cumbersome output normally produced by the procedure.

CONCLUSIONS

This paper uses a simple example to show how macro return codes can be used to conditionally execute parts of a SAS program. This approach can serve as a welcome alternative to the normal output from the procedure and could lead to greater efficiency if coupled with a macro that accepts the return code as a parameter.

REFERENCES

SAS OnlineDoc 9.1.3: <http://support.sas.com/onlinedoc/913/docMainpage.jsp>

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Please don't hesitate to contact the author at:

Robert Adams
Merck and Company, Inc.
UG1D-10
351 North Sumneytown Pike
North Wales, PA 19454
267-305-8249
robert_adams2@merck.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.
Other brand and product names are trademarks of their respective companies.